Some Remarks on Robot Vision

by

Jacob T. Schwartz
and
Micha Sharir

Technical Report No. 119
Robotics Report No. 25

April, 1984

Some Remarks on Robot Vision[1]

Jacob T. Schwartz & Micha Sharir
Courant Institute of Mathematical Sciences
and
School of Mathematical Sciences, Tel Aviv University

I (J.T.S.) share the pleasure of the other speakers at this symposium in paying tribute to Marvin Denicoff. Of the many things that distinguished his career at ONR, the most noteworthy was, I believe, his clear and sustained commitment to fundamental problems of long term significance. The robotics laboratory at NYU, which Marvin founded during one of his last ONR years, reflects this commitment, and we will try to work in a style worthy of Marvin's high goals.

Although the remarks that follow raise more problems than they answer, we hope that they at least suggest some lines of attack along which progress toward the goal of more effective vision systems for robots, a problem of continued and active interest to Marvin, will prove possible.

1. Introduction

The goal of robotics is to develop general-purpose mechanisms having 'operative' intelligence, i.e. that rudimentary level of intelligence which is displayed in the every-day handling of objects in the workplace and the home. For this level of capability to be realized, a robot will need to maintain at least a partial model of its environment internally. Such a model would represent the (known aspects of its) environment in symbolic fashion, as a collection of 'objects' having known shape and orientation. Rigid objects are simplest; however a complete environment model would eventually have to accommodate *flexible* objects like rope, paper and cloth, *liquids*, *soft* objects (e.g. mashed potatoes), *amorphous* objects like dustpiles or heaps of crumbs, etc. Foregoing these interesting but more difficult problems, the following remarks will concentrate on the relatively simple class of rigid objects and on the problem of identifying such objects and determining their orientations so that they can be manipulated by a robot. Of course, manipulation also requires an understanding of object properties and inter-object relationships such as centers of mass, relationships of support, and coefficients of friction, all of which are concepts which a capable robot will have to understand. However, we ignore all these issues to focus on the underlying, still unsolved, problem of how to recognize objects, seen from unknown orientations, and possibly seen as parts of complex multi-object scenes.

---

[1]Work on this paper has been supported in part by Office of Naval Research Grant No. N00014-82-K-0381, and by grants from the US-Israel Binational Science Foundation the Digital Equipment Corporation, the Sloan Foundation, and the System Development Foundation.

Object recognition begins with raw perceptions, i.e. pixel arrays. These must be analyzed into objects which, if rigid, can be identified by the shapes of their bounding surfaces (but may also have properties other than their shape which can be used to identify and locate them, including color, albedo, acoustic reflectance, magnetic behavior, visual texture, electrical behavior; indeed, any property that a sensor can detect). To analyze the objects in a robot's environment will be more or less difficult depending on whether the objects which can appear are known *a priori*, and on whether observation can be maintained continuously or only applied occasionally.

(1) In a highly controlled environment, it may be known that only certain objects, or only objects belonging to known classes, whose members are precisely characterized by a small number of parameters, can be present. If these objects are known to change position only when the robot moves them, and if none of the robot's manipulations miscarry, it may be possible to keep track of the objects, without much sensing, by a kind of 'dead reckoning'. Even if some manipulations miscarry, it may be known (or plausibly assumed) that only certain of the objects will move to unknown positions when an attempted motion fails. It may then be possible to find these objects again by differencing the pre-manipulation scene and its miscarried result, e.g. after dropping coins on a smooth floor one can locate them again by searching visually for shiny raised areas on the floor.

(2) Even if some of the objects move independently of the robot system, it may be possible to maintain a valid environment model by keeping the moving objects under continuous observation. In that case the robot system may be able to follow the position of all objects at all times, and will, e.g., retain the capability of returning them to their base positions on command. For example, a future home robot system with multiple eyes built into the walls of a house might be able to keep all a family's dishes under continuous observation during a meal or party, after the conclusion of which it could return them to their standard positions (after cleaning).

(3) When new objects can appear in the robot's environment, they will at first be perceived simply as unexpected surfaces. It will then be necessary to analyze these surfaces into the objects for which they belong. This process may be able to exploit *a priori* knowledge that only objects of certain categories will ordinarily appear in the robot's environment.

To be fully useful, the recognition capabilities described in the preceding pages need to be organized appropriately. What is wanted is basically a pair of procedures. The first of these should be an object acquisition procedure to which a succession of objects can be presented and their identities supplied. The acquisition procedure should record the object shapes in some suitably compressed form and can pre-process these shapes so as to obtain a collection of efficient

discriminating tests for subsequent object identification. The second procedure will then use this data to ingest a scene containing one or more of these objects and convert it into a list of the objects present, each with its identity and orientation.

Going one step further, we can describe the goal of vision-based object recognition system as follows. The system should maintain and continually update a set of 'registers', one for each object observed; each of these registers should at all times contain the position and orientation of the corresponding object. For moving objects, the system should update this information continuously. This implies that after initial scene analysis the system must frequently probe to determine how the objects in the scene have moved, and whether new objects have entered the scene.

The values present in such continuously updated registers can be considered to represent the natural outputs of an advanced robot vision capability since they are just what the system needs to control and manipulate its environment. Practical progress in scene analysis will be defined by the classes of 2-D and 3-D objects for which we are able to make such a symbolic interface to the real world available and reliable.

2. Advantageous Forms of Raw Data

Visual information is most useful if it is given as 3-D visual data, i.e. if true 3-D coordinates are immediately calculated for all points observed. We note that devices which provide such 3-D data, usually based on laser-beam scanning or on use of specially 'structured light' are commercially available already; see e.g. [S79], [Sc83], [T83].

The crucial advantage of 3-D vision is that it allows images to be acquired by arbitrarily many eyes. Whereas to take ordinary (2-D) images acquired by several eyes and combine them is not easy, multiple 3-D images of a single scene combine in a trivial way, since they all refer to surfaces in a common geometric space. This makes it possible to use arbitrarily many eyes, some fixed, others mounted on moving parts of the robot system. (Eyes need to be mounted on the robot itself if either the robot can roam freely, or to ensure that the space near moving portions of the robot is not obscured, either by an intervening object or by parts of the robot itself. This second purpose may require specialized eyes of appropriate form and position.) Note that an 'all seeing' eye system of this sophistication subsumes a quite satisfactory proximity sensor, and makes other forms of proximity sensors superfluous.

Of course, this still leaves open technical questions such as: how to combine separate observations; what to do when surfaces seen by more than one eye differ discernably; and when to reject an interpretation because of unacceptably large discrepancies. Nevertheless 3-D images are basically favorable for combination, while 2-D images are basically much less favorable.

C.1

3. Image Interpretation Techniques

The visual data gathered by a 3-D sensor, i.e. '3-D' or 'depth' images, can be grouped in a table listing all points in 3-space which lie on some reflecting surface of one of the objects present in a scene. However, since all sensor-gathered data is partly corrupted by noise, acquisition of 3-D images leads at once to the problem of how to identify objects, given slightly noise-corrupted images (or, in other cases, silhouettes) of them.

Whether depth images or silhouettes are in question, we shall assume that the objects present in the scene to be analyzed are drawn from some known collection of possible objects $O_1, \ldots, O_n$. This assumption makes the image analysis problem 'objective' rather than 'psychological': one just wants the computer looking at a scene to calculate an integer or a finite set of integers that tells us exactly which of a known list of possible candidate objects it sees, and from what angles it sees them. However, our simplifying assumption still leaves us free to consider any one of a scale of image interpretation problems of gradually increasing difficulty, all of which are 'objective' in the sense just mentioned, and all of which would contribute robot capabilities of practical significance if solved:

(1) The bodies present in the scene can be wholly visible or may be partially obscured.

(2) The bodies can be straight-sided (polygonal or polyhedral), or curved.

(3) If the bodies present in the scene are polygonal or polyhedral, they may either be known to lie in some constrained orientation, (e.g. standing on an edge or face, atop a flat surface), or can be present in completely arbitrary orientations.

(4) We may be able to assume that the scene contains just one object, or may need to deal with scenes containing multiple objects, which may have either a single uniform orientation, or many different orientations.

(5) Instead of fixed objects, we may need to deal with objects which are known only to belong to one of a finite sequence of object classes $O_1(s), \ldots, O_n(s)$, each of which depends on one or more shape parameters s (e.g. in a home robotics application, cylindrical cans of various heights and widths may be encountered).

This list defines a family of problems for whose solution appropriate algorithmic or heuristic approaches are needed. The efficiency of the approach selected will be important, especially in the dynamic case where the system needs to keep track of moving objects in real time.

The remarks which follow will describe various semi-algorithmic heuristics of gradually increasing complexity which can be used to handle some of the problems listed above. Some of these approaches have been simulated, and where possible we will note the results of simple numerical experiments. The approach proposed is related to one explored in a series of papers by Y. Shirai of the Tsukuba

Electotechnical Laboratory in a series of papers; see [OS75], [OS79], [S79], [SKOI83], [SS71].

4. Recognition of 2-D Objects

Two basic approaches to recognition of 2-D objects drawn from a finite collection of candidate objects can be proposed. The first approach applies successive 'probes' to the object, gathering sufficient information to discriminate it from among other potential objects (and to determine its orientation). This approach deals particularly easily with simple polygonal objects, but sidesteps the issue of shape description. The second approach associates a global shape descriptor with each object viewed, and then matches this descriptor to pre-calculated similar descriptors of the model objects expected to be seen.

Recognition by Probing

By processing raw image data in simple ways we can apply various logical 'probes' to it. (In the absence of visual data, probing can be accomplished mechanically by detecting object contact using touch sensors.) Each such probe can be considered to move along some specified curve γ from a given position in a given direction until the first intersection of γ with an object present in the scene is detected. The curves along which one probes can be straight lines, circles, etc. If only silhouettes of an object are given, we can still think of 'silhouette probes', i.e. probes in the silhouette plane which end on encountering a point of the silhouette boundary.

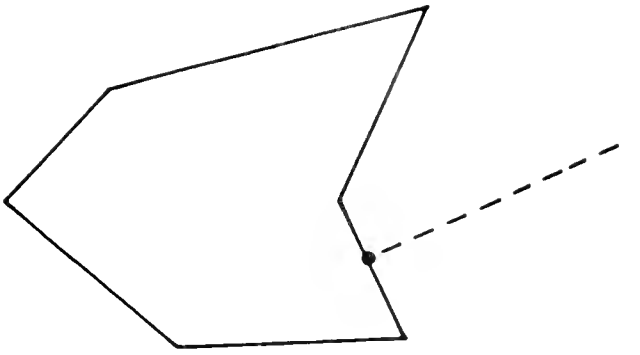

Figure 1(a): Probe of a 2-D polygonal object

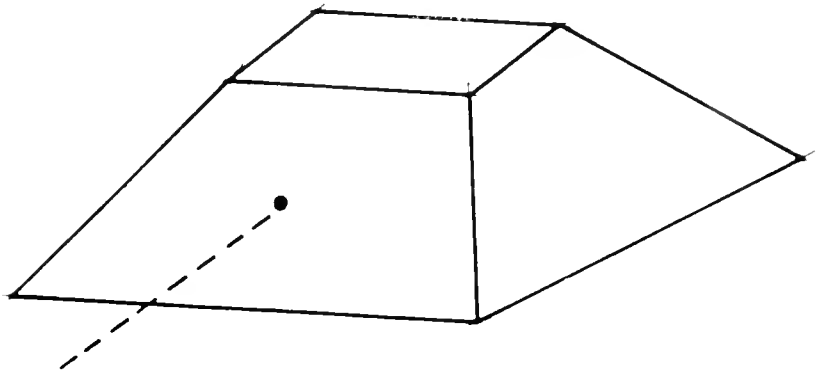

Figure 1(b): Depth probe of a 3-D object

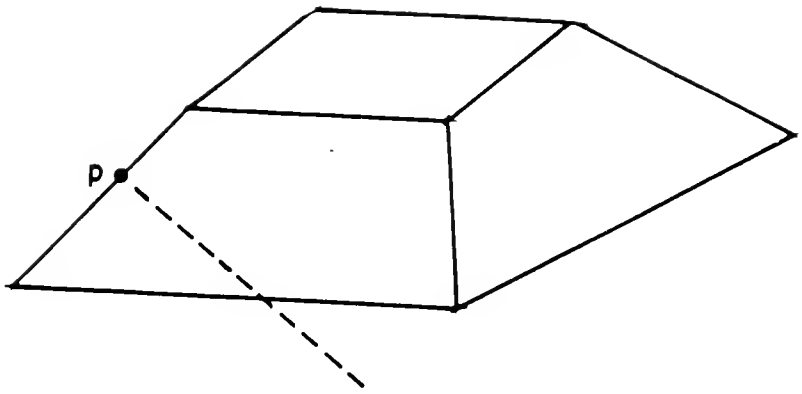

Figure 1(c): Silhouette probe of a 3-D object
(Distance from camera of point p need not be known.)

To see how easily objects can sometimes be identified by probing, consider the simplest 2-D case, in which we are given a polygon standing on one of its edges. In this particularly elementary situation, the first probe conducted will establish a point against which the polygon can be considered to 'fit', and then, knowing the point at which the probe contacts the polygon, we know that the polygon's possible positions are restricted to a finite set. This allows us to build the finite set of points at which a further probe line could intersect one of the polygons which might confront us, in one of its finitely many possible orientations. Suppose we divide this probe line into minimal resolvable intervals (determined by the precision of the instrument with which probes are conducted). Count the number of such intervals which contain points of intersection and calculate the entropy of the associated subdivision; this is the *resolving power* of the probe line. For efficiency one will then want to probe along the line whose resolving power is greatest. Only a subset of the original set of polygons and orientations will remain as candidates after this first probe, and then one can apply a second probe which has greatest resolving power for this subset, etc. The tree-like search which results will determine the identity of the observed polygon and the edge that it is standing on. Normally very few probes will be required. The first probe should be at a level which minimizes the expected number of probes subsequently required. (This style of searching associates a notion of 'entropy', relative to the imprecision of the probing instrument, with the given set $O_1, \ldots, O_k$ of objects; this 'entropy' is likely to have interesting invariance properties, and deserves closer study.)

Note that the probing procedure outlined is independent of any assumption of convexity.

Next consider the somewhat less trivial case of a convex polygon whose initial orientation is totally unknown (but assume that we know one point interior to it). Probe the polygon twice to determine two points on its periphery, and then track the segment between them to determine whether these two points belong to the same polygon edge (we assume that such a 'generalized probing' operation is available). If not, probe at a point intermediate between the two first points, and repeat. Eventually we must find two points which lie together on the same edge of the polygon; this reduces us to the case considered previously, since the polygon may be considered to be 'standing' on this edge.

Similar ideas can be applied to the more interesting case of a curved 2-D region. To avoid complications suppose first that the region is convex. If we can locate one point P fixed relative to the region, then, by probing along a circle about this point as center, we can orient the region so that a chord through the point P having a standard length D can be regarded as horizontal. This standardizes the position of the region to one of a finite collection of possible positions, and then we can use the kind of 'probe tree' described above to determine which one of these possible orientations it has.

If the whole of a region is visible, its centroid can serve as such an anchor point. Similar use could also be made of the two most distant points of the region, of the point of the region most distant from the line connecting these two points, etc. Polygon corners can obviously be used as anchor points; acute corners, of which only a few can exist, are obviously preferable to obtuse corners. It is only necessary that any point which probing might identify with a particular anchor point P should belong to some relatively small known set of points fixed relative to the region, but when the anchor point is ambiguous, the probe tree which we build up must reflect all the possible points that might be confused with it.

Next consider the case of partially-obscured 2-D objects. The preceding observations suggest that the first step in recognizing partially obscured objects should be to define noise-immune anchor points which can be located even if part of the region is obscured. This case is of course more difficult than that in which the whole object is visible, because

(1) For totally visible objects, obvious 'global' anchor points such as the object centroid are available, while for partially occluded objects anchor points must be calculated from relatively 'local' data.

(2) If the object being observed is nonconvex, then the (boundary of the) convex hull of the portion being observed need not lie on the convex hull of the whole object (see Fig. 11).

Of course, if the observed portion of the object has sharp discriminating features such as acute corners, then finding an anchor point will be relatively easy. If no such sharp features are present, the problem becomes more difficult. One way of approaching it is by associating the object with some appropriate,

geometrically defined function on the unit circle/sphere of directions in 2-D (resp. 3-D) space. The function must be one whose geometric definition makes it invariant with respect to Euclidean motions of the region to be analyzed. Whenever such an artificial 'color' shows sharp transitions or peaks, these can be used to define the anchor points that we need. In effect, this notion of 'artificial color' converts the shape recognition problem into the problem of recognizing 'colored beachballs' when these are seen from an unknown orientation, a problem for which the presence of spots or regions of sharply defined color will clearly be significant.

Artificial colors of the type proposed can be defined in very many ways, but we want to chose one which has peaks or which varies sharply in the vicinity of geometrically significant boundary features of the body to be analyzed. One possible scheme is as follows: take a modified "carpenter's square" MCS, consisting of two half-lines making some standard angle α ($\alpha = 90°$ would be the standard carpenter's square) and fit it over the region so that both of its two sides touch the boundary of the region. The point at which this contact occurs is determined by the orientation θ of (some distinguished one of) the sides of MCS.

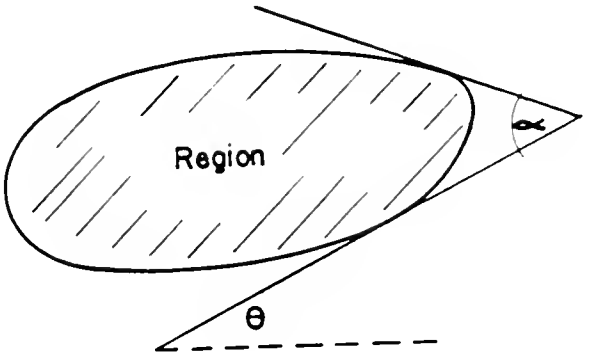

Figure 2: Modified 'carpenter's square' in contact with two points of a body. 'Leading' side is at angle θ to the horizontal.

Let A be the apex of the modified carpenter's square MCS. Take the segment connecting the two points of contact between the region and MCS, take the midpoint A and M of this segment, and then find and record the distance $d(\theta) = d(\theta,\alpha)$ between the point x at which the line from A to M crosses the region boundary.

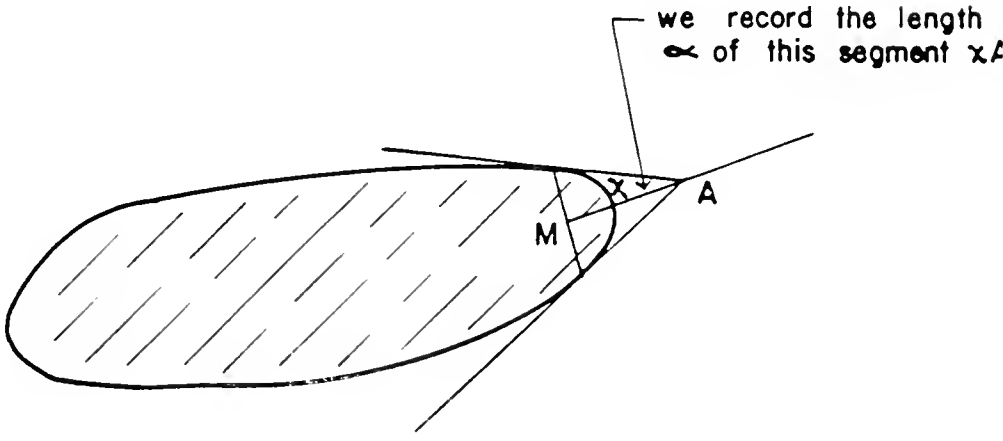

Figure 3: Using a modified carpenter's square to measure a region's average boundary curvature.

If the region were simply a circle of radius R, then the distance $d(\theta)=d(\theta,\alpha)$ would be independent of θ, and would in fact be $R(\sin\alpha/2)^{-1}-1$. Thus $d(\theta)$ measures a kind of average of the curvature of the periphery of the region; averaged, that is, over the section of periphery between its the region's two points of contact with MCS.

The angle α that can be used in measuring the periphery of a partially obscured region depends on what portion of the periphery is visible. To use an angle α, the tangent to the visible portion of the periphery must turn through an angle exceeding $180° - \alpha$. The closer α approaches 180°, the closer $(\sin\alpha/2)^{-1}-1$ comes to 0, and hence the more sensitive $d(\theta)$ becomes to small measurement inaccuracies.

If $d(\theta)$ is constant (for several values of the apex angle α of our modified carpenter's square MCS), then the region (or rather the visible portion of its periphery) must be circular, and hence actually possesses no geometric features other than its radius. If $d(\theta)$ is nearly constant, i.e. if the ratio of its largest to its smallest values lies near 1, then the (visible part of the) region will be nearly circular, and hence relatively featureless geometrically. Otherwise this ratio will vary more substantially, enabling us to locate anchor points relatively sharply. To expand upon this remark, it is convenient to consider not $d(\theta)$ but its logarithm $D(\theta) =$

log d(θ). By assumption, D(θ) varies substantially from its minimum value (over the visible part of the periphery, which correspnds to a range of angles <= 2 π). Suppose that the smallest change in D that we feel able to measure is a change ϵ. Establish a succession of levels δ,δ+ϵ,δ+2ϵ, ... through the range of D. For each of these levels δ_i, divide the range over which θ varies into disjoint intervals, each containing a point at which D takes on the value δ_i, and each terminated by the first occurrence of a sufficiently large interval in which D dips below $\delta_i-\epsilon$ or rises above $\delta_i+\epsilon$. Choose one representative point in each such interval, take this as an anchor point, and make corresponding entries in a probe tree.

To identify a region using this information, we can subsequently survey it with a generalized carpenter's square of appropriate apex angle (depending on the amount of unobscured periphery available, which is appropriately measured in terms of the number of degrees through which the periphery has turned.) Once having measured the boundary in this way, find intervals as above; that is, intervals each of which contains at least one point θ for which $D(\theta)=\delta_i$ and terminated in the same way as the intervals used to build the probe tree. Examine these intervals for each level δ_i, and take the smallest; this gives the most definite information concerning the location of the corresponding anchor point. Then divide this interval of orientations into subintervals, each small enough so that no point of intersection with a probe line can move by more than the standard measurement uncertainty of a probe when the object turns through a single orientation step. In effect, this rule defines the number of 'micro-facets' into which our procedure must divide the interval.

Moving through this range of orientations by stepping successively between the intervals into which we have divided it, take the point x of Fig. 3, which lies between the point M and the apex A of the GCS, as an anchor point, and then execute (or simulate) a series of probes. This will eliminate incorrect orientations/identifications, normally quite rapidly, and leave only those orientations consistent with the available data concerning the visible periphery.

Note that the number of orientations over which we need to search serially will be roughly proportional to

$$\min |I| / \mathrm{var}\,(D|I)$$

where I designates an angular subinterval of the visible range of tangent angles (to the region periphery), | I | is the size of this subinterval, and D | I designates the restriction of the function D to the subinterval I. Thus favorable cases are those in which a substantial part of the variation of D takes place in some small range of angles; unfavorable cases are those in which D varies uniformly over the whole of the visible angular range. Even in this unfavorable case, the range of angles we have to search will be limited to a fraction of the total angular range inversely proportional to D's total variation.

The following additional technique can be used to improve the efficiency of the simple approach just outlined. Suppose that the function(s) D using which we are trying to identify and orient a region are constant or nearly constant over some substantial portion B of a region boundary. Then this section B of the boundary is likely to be close to circular, and of a known radius R. We can exploit this fact by mapping the visible portion of the boundary to a much smaller curve. This can be done by moving each of its points P a known distance d (easily calculated from the estimated radius R) perpendicularly away from the tangent line at p. The image of B is then a significantly smaller curve B'. (The image of a perfect circle would plainly be the unique point fixed relative to the circle, i.e. its center.)

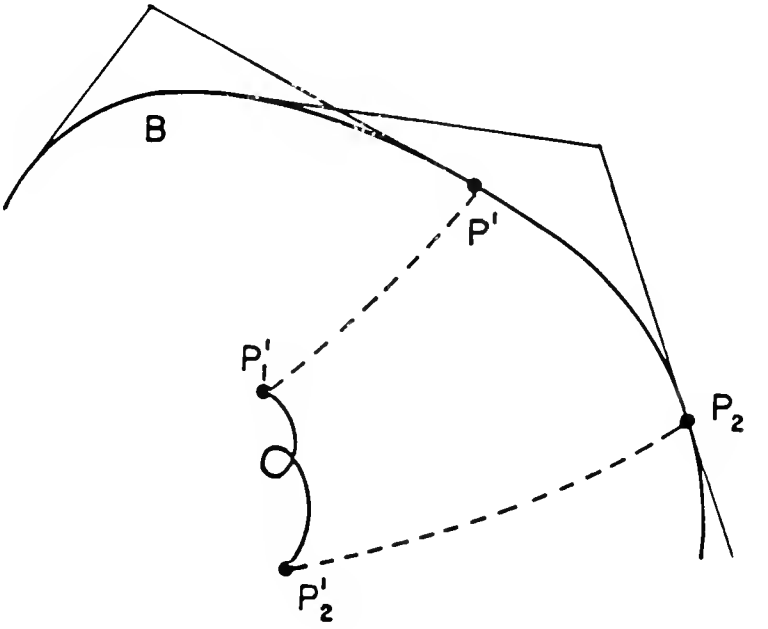

Figure 4: Mapping a nearly circular curve into a smaller curve by 'radial' translation of its points.

Once B has been constructed, we can cover it with sufficiently small circles, which, hopefully, will not be very numerous. The centers of these circles form a collection A of possible anchor points, in the sense that when the curve is measured (with a generalized carpenter's square) and found to have a D-value which only limits the region orientation to the large angular interval B, any point constructed in the manner just explained must lie very close to one of the anchor

points in A.)

5. Some observations concerning geometrically 'colored' and 'colorless' curves and surfaces in 2 and 3 dimensions

The issue crucial to some of the region identification techniques outlined above is how to find one or more 'anchor points' which can be used to standardize the position of the region. (Similarly, flat sides of a region define 'anchor orientations'.) Once such an anchor point has been found, the identification problem becomes very much easier. An anchor point may be unique, or, as in the case of a polygon, many possible points (vertices) may define useful anchors. Moreover, anchor points may be uniquely identified by geometric invariants associated with them, or, as in the case of a regular polygon, a region may possess symmetries and therefore possess multiple anchor points which fall into logically indistinguishable categories.

As we have noted, as soon as a curve is 'painted' with some concrete or abstract 'color' which has significant variation along the curve, it becomes is easy to define anchor points; it suffices to take those points having some characteristic color, (but for this we want to pick a color which occurs only infrequently on the curve.) The rotational invariants occuring in the preceding discussion give us a way of operating in situations in which no external color is available, by forming noise-immune geometric invariants and using them as generalized colors. (These 'geometric colors' are most naturally associated with orientations of a measuring instrument and thus can most naturally be regarded as painting the circle rather than the region boundary under investigation. Since, in the case of convex bodies, each orientation maps naturally to a point of the region boundary, this viewpoint loses no significant information.)

We can best understand the potential of this approach by considering those situations in which it must fail. These are situations in which the boundary curve being measured is completely 'colorless' relative to the geometric invariant calculated, i.e. cases in which the battery of invariants we bring to bear have constant values over the boundary of the object being measured. Note that these are also cases other shape matching techniques will also tend to fail, because the same degree of matching will be attained by a large family of orientations differing simply by Euclidean motions, making it impossible to discriminate between these orientations.

To be satisfied with a collection of geometric invariants, we will want constancy of the geometric invariants used to 'paint' an object's boundary to imply that the boundary is *inherently colorless* geometrically, i.e. to imply that its points are equivalent to each other under a Euclidean motion of the whole plane. Curves having this property must clearly be orbits of points under 1-parameter subgroups of the group of plane motions, and hence must either be circles or straight lines (note therefore that if we can see the whole boundary of an object, the circle is the only possible colorless curve). Let us call a set of geometric

invariants *ample* if any curve for which these invariants are constant over the length of a curve is necessarily straight or circular. (In addition, we want invariants that are *stable* relative to small perturbations of a curve, and which are *local*, allowing them to be calculated for nearly the full angular range through which a partially obscured convex curve turns.) Once we have an ample set of invariants (also possessing the other properties just noted) we will have done as well as we can, in the sense that invariants better in any ideal sense are impossible. Similar considerations apply to curves in 3-space and to curves which lie in other geometric objects of concern to us, particularly curves on the sphere.

A similar of geometric 'colorlessness' applies to curves in 3-space and to surfaces. A geometrically colorless curve in 3-space is either a straight line, circle, or helix. A colorless curve lying on the surface of the sphere is necessarily a circle (not necessarily a great circle). A similar notion and remark apply to colorings of the sphere; such a coloring fixes a point (which can then be used as an anchor point) unless there exists a continuous group of rotations of the sphere which leaves the coloring invariant, i.e. $c(Rp) = c(p)$ for the color (or colors) c and every R in some continuous group of rotations. Here there are only two possibilities: either c is constant, or c is constant on each of a family of parallel circles on the sphere. In all other cases, either changes in the shape of one of the level curves $c(p) = const$ will fix a point, or changes in the relative position of two level curves fix such a point. (For example, for each point p on a first (circular) level curve $c(p) = const_1$ we can take its minimum distance to a second (also circular) level curve $c(p) = const_2$; unless the two curves are parallel, this function paints a varying geometric 'color' along the first curve, and (assuming infinite precision) this color fixes an anchor point.

Next consider surfaces in three dimensional space which are geometrically colorless, either in the strong sense that all their points are geometrically equivalent, or in the weaker sense that the surface is invariant under some one-dimensional continuous subgroup of the Euclidean group. In the first case, the surface must have constant principal curvatures, and hence must be a portion either of a plane, sphere, or circular cylinder. In the second case, the orbit of any point under the group of motions leaving the surface invariant must be either a straight line, circle, or helix (of pitch determined by the group leaving the surface invariant). Hence the surface must be either a portion of a cylinder (not necessarily circular), a surface of rotation, or a 'helical cylinder' (screw surface) defined by its cross-section in a plane perpendicular to the direction of the common helix axis.

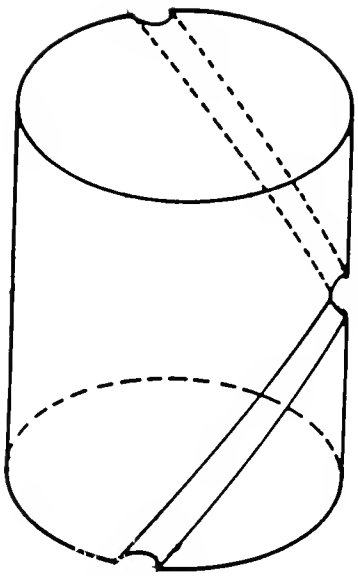

Figure 5: Example of a 'helical cylinder'.
All points of each helix on the surface have equal color.

While interesting as mathematical examples, helical cylinders of this kind are (except for machine screws and their colored equivalent, barber poles) rare.

6. Shape Descriptor Matching

Probing methods like that described above use the actual periphery of the region to be identified, and do not embody any global concept of region shape. This contrasts with other identification techniques that work from some abbreviated shape descriptor which can be associated with the periphery of a convex region, rather than from the periphery itself. The stability and efficiency with which these descriptors can be matched is crucial for such methods. A few mathematical observations can be made concerning this point. Assume first that the observed region is expected to be convex. Such regions are often described by their 'turning function' $\theta(s)$, i.e. the function which, starting from some arbitrarily designated point of its periphery and proceeding counterclockwise around the periphery, records the change in angle of the counterclockwise tangent as a function of the arc-length s traversed. This function is monotone increasing, and varies through 2π as s goes from 0 to its final value S, which is the total periphery

of the region. The value S simply describes the total size of the region, and (if the whole periphery is available) we can normalize it to 2π, so that $\theta(s)$ is monotone and goes from (0,0) to $(2\pi,2\pi)$.

The function $\theta(s)$ has various useful properties:

(1) $\theta(s)$ is invariant under any Euclidean motion of the object O in question.

(2) $\theta(s)$ depends in a very simple way on the starting point on the boundary of O, that is, if the starting point shifts by s_0, the graph of θ undergoes a correponding horizontal and vertical shift, i.e. simply changes to

$$\theta'(s) = \theta(s + s_0) - \theta(s_0)$$

We can still use the shape descriptor $\theta(s)$, measured along the visible portion of O's boundary, even if O is partially occluded, i.e. even if only the portion of O which lies right of some (known) directed line is visible. In such case the graph of θ will simply be an (appropriately shifted) portion of the graph for the whole boundary of O.

(3) θ is parametrized by the arc length of the boundary of O, which can become unstable under small perturbations if convexity is lost. That is, if we represent the noise-corrupted boundary of an observed object O simply as the polygonal line passing through all observed boundary points, the resulting arc length can differ greatly from that of the ideal, noise-free object, in which case the shape descriptors for the observed body and for its model counterpart will not approximate one another. To overcome this difficulty, we can compute the convex hull of the observed data points, obtaining a convexified observed object, and then match the shape descriptor for this convexified observation to pre-stored data describing various model convex bodies.

If the region is polygonal, the function $\theta(s)$ is a step function whose discontinuities tend to be troublesome when a slightly perturbed measurement of $\theta(s)$ is matched against a pre-stored model. To avoid this problem, one can simply turn the graph of the function 45° clockwise, thereby converting it into the graph of a revised function $\eta(s)$ whose derivative is bounded by 1 in absolute value.

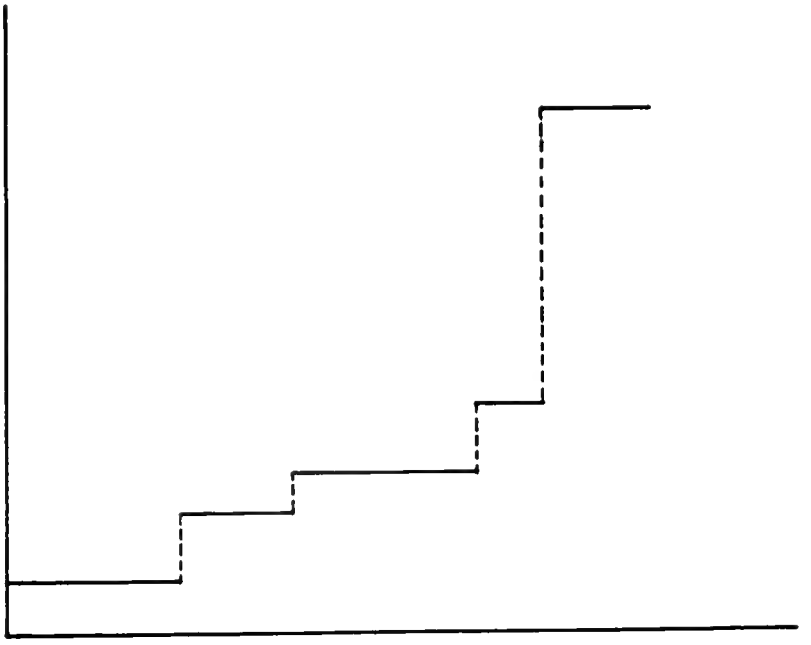

Figure 6(a): Graph of $\theta(s)$ for a polygon.

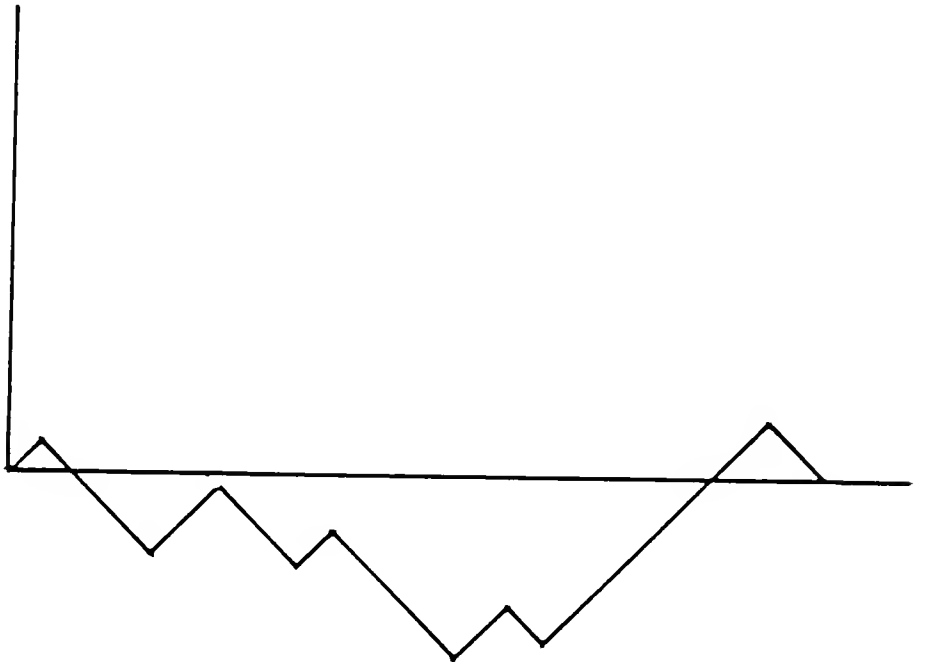

Figure 6(b): Graph of $\theta(s)$ after rotation by 45°.

To identify an observed convex region O (which can be partially obscured), we can then compute its descriptor $\theta(s)$ and try to match it to similar shape descriptors computed for model regions $O_1, \ldots, O_n$ corresponding to the various 2-D objects which might appear in an observed scene.

In situations of this kind, matching is customarily implemented by means of the fast Fourier transform algorithm. More specifically, let $\xi(s)$ denote the graph of boundary turning-angle vs. arc length computed for a (convexified) observed polygon O after this graph is turned clockwise by 45° (if O is obscured by a line l then the graph of ξ should start at one point of obscuration (i.e. one intersection of l with the boundary of O) and end at the other such point). For each of the specified model objects O_j let η_j denote the corresponding (rotated) graph for O_j.

If we can use the L^2 metric in shape descriptor space, the object O_j matching O most closely is that which minimizes the distance

$$(2) \qquad \min_{d_0} \int_0^L |\eta_j(x+d_0) - \eta_j(d_0) - \xi(x)|^2 dx$$

Actually, we find it better to change (2) slightly so as to find the best "vertical fit" between $\eta_j(x+d_0)$ and $\xi(x)$, i.e. to minimize

$$(3) \qquad \min_{d_0} \min_{c} \int_0^L |\eta_j(x+d_0) - \xi(x) - c|^2 dx$$

In (3) the best value of c is given by

$$(4) \qquad c = c(d_0) = \frac{1}{L} \int_0^L (\eta_j(x+d_0) - \xi(x)) dx \, .$$

With this value of c, (3) becomes

$$(5) \qquad \min_{d_0} \left(\int_0^L |\eta_j(x+d_0) - \xi(x)|^2 dx - L|c(d_0)|^2 \right) =$$

$$\min_{d_0} \left(\int_{d_0}^{d_0+L} |\eta_j(x)|^2 dx + \int_0^L |\xi(x)|^2 dx - 2\int \eta_j(x+d_0)\xi(x) dx - L|c(d_0)|^2 \right),$$

where in the third integral in the last form of (5) we take ξ to be defined as zero outside the interval $[0,L]$. This allows the minimum appearing in (5) to be rewritten as

$$(6) \qquad \min_{d_0} \Big(I_j(d_0 + L) - I_j(d_0) - \frac{1}{L} \Big[K_j(d_0 + L) - K_j(d_0) - \int_0^L \xi(x) dx \Big]^2$$

$$+ \int_0^L |\xi(x)|^2 dx - 2\int \eta_j(x+d_0)\xi(x) dx \Big) \, ,$$

where

(7) $$I_j(d) = \int_0^d |\eta_j(x)|^2 dx$$

$$K_j(d) = \int_0^d \eta_j(x) dx \ .$$

Since the most expensive part of the computation (6) is simply a convolution, it follows that, after discretization to n interpolating points, we can calculate the minimum (6) (for each j separately) in time $O(n \log n)$, using the fast Fourier transform technique.

Direct Match of Rotated 2-D Objects

A cruder but stabler and still quite effective 2-D shape matching scheme can also be implemented efficiently using the fast Fourier transform. In this method, we simply take a sequence of points equally spaced along the perimeter of a (convexified) observed polygon O. More precisely, we take a sequence $(u_1, \ldots, u_n)$ of points in clockwise order along the boundary of the convex hull of O such that all the arcs between successive points u_i and u_{i+1} have equal lengths (which must therefore be S/n, where S is the total length of the periphery of O). We then wish to match two such sequences $(\mathbf{u}_j)_{j=1}^n$ and $(\mathbf{v}_j)_{j=1}^n$ corresponding to an observed (convexified) object O and a model object M respectively. Assume first that the whole boundary of O is visible. Matching amounts to finding a Euclidean motion E of the plane which will minimize the L_2 distance between the sequences $(E\mathbf{u}_j)_{j=1}^n$ and $(\mathbf{v}_j)_{j=1}^n$; i.e. we need to compute

$$\Delta = \min_E \sum_{j=1}^n |E\mathbf{u}_j - \mathbf{v}_j|^2$$

To simplify this calculation, first translate O so that its centroid lies at the origin, giving

$$\sum_{j=1}^n \mathbf{u}_j = 0$$

Next write E as $E\mathbf{u} = R_\theta \mathbf{u} + \mathbf{a}$, R_θ denoting a counterclockwise rotation by θ. Then

$$\Delta = \min_{\theta, \mathbf{a}} \sum_{j=1}^n |R_\theta \mathbf{u}_j + \mathbf{a} - \mathbf{v}_j|^2 =$$

$$\min_{\theta, \mathbf{a}} \left[\sum_{j=1}^n |\mathbf{v}_j|^2 + n|\mathbf{a}|^2 - 2\sum_{j=1}^n \mathbf{a}\cdot\mathbf{v}_j + \sum_{j=1}^n |\mathbf{u}_j|^2 + 2\sum_{j=1}^n \mathbf{a}\cdot R_\theta \mathbf{u}_j - 2\sum_{j=1}^n R_\theta \mathbf{u}_j \cdot \mathbf{v}_j \right]$$

But

$$\sum \mathbf{a}\cdot R_\theta \mathbf{u}_j = \mathbf{a}\cdot R_\theta(\sum \mathbf{u}_j) = 0 \ .$$

Hence $\mathbf{a}$ and θ appear independently in Δ and we can minimize their contributions separately.

To minimize over $\mathbf{a}$ simply put

$$\mathbf{a} = \frac{1}{n}\sum_{j=1}^{n} \mathbf{v}_j$$

As to θ, we need to compute

$$\delta = \max_{\theta} \sum_{j=1}^{n} R_\theta \mathbf{u}_j \cdot \mathbf{v}_j$$

Regarding the vectors $\mathbf{u}_j$, $\mathbf{v}_j$ as complex numbers u_j, v_j, we can rewrite this as

$$\delta = \max_{\theta} \operatorname{Re}\left[\sum_{j=1}^{n} e^{i\theta} u_j \bar{v}_j\right] = \left|\sum_{j=1}^{n} u_j \bar{v}_j\right|$$

Altogether this gives

$$\Delta = \sum_{j=1}^{n} |\mathbf{v}_j|^2 - \frac{1}{n}\left|\sum_{j=1}^{n} \mathbf{v}_j\right|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2\left|\sum_{j=1}^{n} u_j \bar{v}_j\right| \qquad (*)$$

$$\Delta = \sum_{j=1}^{n} |\mathbf{v}_j|^2 - \frac{1}{n}\left|\sum_{j=1}^{n} \mathbf{v}_j\right|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2\left(\left|\sum_{j=1}^{n} \mathbf{u}_j \cdot \mathbf{v}_j\right|^2 + \left|\sum_{j=1}^{n} \mathbf{u}_j \times \mathbf{v}_j\right|^2\right)^{\frac{1}{2}},$$

where $\mathbf{u}\times\mathbf{v}$ denotes the (2-dimensional) cross product of the vectors $\mathbf{u}$ and $\mathbf{v}$. (Note the similarity between (*) and the formula for the best matching between two turning-angle shape descriptors given in the preceding section.)

If O is partially occluded or appears in an unknown orientation, we have to match the sequence $(\mathbf{u}_j)_{j=1}^{n}$ to each of the contiguous subsequences $(\mathbf{v}_{j+d})_{j=1}^{n}$ of the (circular) sequence $(\mathbf{v}_j)_{j=1}^{m}$, for $d = 0, \ldots, m-1$. (We assume that $m \geq n$, for otherwise the (partial) periphery of O is too long to match M.)

For each such d (*) becomes

$$\Delta(d) = \sum_{j=d+1}^{d+n} |\mathbf{v}_j|^2 - \frac{1}{n}\left|\sum_{j=d+1}^{d+n} \mathbf{v}_j\right|^2 + \sum_{j=1}^{n} |\mathbf{u}_j|^2 - 2\left|\sum_{j=1}^{n} u_j \bar{v}_{j+d}\right|$$

As in the preceding analysis, the minimum of the values $\Delta(d)$, $d = 0, \ldots, m-1$, can be found in time $O(m \log m)$, using the fast Fourier transform.

It is interesting to note that the observations made in the last few pages generalize easily to curves in three dimensions, or, more generally, to any situation in which a model curve or surface $\mathbf{u}(\omega)$ depending on one or more parameters ω must be rotated and translated to match a model curve or surface $\mathbf{v}(\omega)$ as well as possible. We need to assume, however, that the matching operation involves no change in parametrization for either of the functions $\mathbf{u}(\omega)$ or $\mathbf{v}(\omega)$.

Suppose more specifically that we are given two descriptor functions $\mathbf{u}(\omega)$, $\mathbf{v}(\omega)$, $\omega \in S$, corresponding respectively to an observed object O and a model object M. We need to find the Euclidean motion E (e.g. of 3-space) which minimizes

$$\Delta = \min_{E} \int_{S} |E\mathbf{u}(\omega) - \mathbf{v}(\omega)|^2 \, d\omega$$

As in the 2-D case, we translate O so that its centroid lies at the origin, giving

$$\int_{S} \mathbf{u}(\omega) \, d\omega = 0$$

Write E as $E\mathbf{u} = R\mathbf{u} + \mathbf{a}$, where R is a rotation. Then

$$\Delta = \min_{R,\,\mathbf{a}} \int_{S} |R\mathbf{u}(\omega) + \mathbf{a} - \mathbf{v}(\omega)|^2 \, d\omega =$$

$$\min_{R,\,\mathbf{a}} \left[\int_{S} |\mathbf{v}(\omega)|^2 \, d\omega + |S||\mathbf{a}|^2 - 2 \int_{S} \mathbf{a}\cdot\mathbf{v}(\omega) \, d\omega + \int_{S} |\mathbf{u}(\omega)|^2 \, d\omega + 2\int_{S} \mathbf{a}\cdot R\mathbf{u}(\omega) \, d\omega - 2 \int_{S} R\mathbf{u}(\omega)\cdot\mathbf{v}(\omega) \, d\omega \right.$$

But

$$\int_{S} \mathbf{a}\cdot R\mathbf{u}(\omega) \, d\omega = \mathbf{a}\cdot R\left(\int_{S} \mathbf{u}(\omega) \, d\omega\right) = 0 .$$

Hence $\mathbf{a}$ and R appear independently in Δ and we can minimize their contributions separately.

To minimize over $\mathbf{a}$ simply put

$$\mathbf{a} = \frac{1}{|S|} \int_{S} \mathbf{v}(\omega) \, d\omega$$

As to R, we need to compute

$$\delta = \max_{R} \int_{S} R\mathbf{u}(\omega)\cdot\mathbf{v}(\omega) \, d\omega$$

To find δ, first calculate the matrix A given by

$$A_{ij} = \int_{S} \mathbf{u}_i(\omega)\mathbf{v}_j(\omega) \, d\omega ,$$

(where $i,j = 1,2,3$ if we are dealing with a curve or surface in 3-space). In terms of the matrix A we can express δ as

$$\delta = \max_{R} \; tr(RA)$$

To maximize $tr(RA)$, decompose A as $A = QH$, where $Q = A(A^{*}A)^{-\frac{1}{2}}$ is a pure rotation and $H = (A^{*}A)^{\frac{1}{2}}$ is positive definite symmetric. This gives

$$\delta = \max_{R} \; tr(RQH) = \max_{R} \; tr(RH) = tr(H) = tr((A^{*}A)^{\frac{1}{2}})$$

To see this, note that since the trace is invariant under rotation, we can assume

that H is diagonal. But for a diagonal positive definite matrix (λ_i) and a rotation matrix (r_{ij}) the trace of the product $\sum \lambda_i r_{ii}$ can be no larger than $\sum \lambda_i$ and can assume this value only when $r_{ij} = \delta_{ij}$.

Overall, we have

$$\Delta = \int_S |v(\omega)|^2 \, d\omega - \frac{1}{|S|} \left| \int_S v(\omega) \, d\omega \right|^2 + \int_S |u(\omega)|^2 \, d\omega - 2 \, tr((A^* A)^{\frac{1}{2}}) \qquad (**)$$

showing that the optimal rotated match between O and M can be found in time proportional to that needed to integrate the various functions appearing in (**), i.e. proportional to the number of data points used to discretize the curves or surfaces u and v.

Much as in the 2-D case considered previously, these formulae can be used to match observed 3-D curves parametrized by arc length to similarly parametrized model curves. Matching can be achieved in time $O(n \log n)$ by using the fast Fourier transform, even if the observed curve O is partially obscured. This remark is potentially applicable to matching of 'iso-color' curves on 3-dimensional surfaces.

There are two difficulties in extending the matching technique just described to partially obscured surfaces. The first difficulty is to parametrize O. This point is discussed below; but the obvious parametrization using the centroid that can be used in the unobscured case is not available for partially obscured objects.

A second difficulty involves the computational cost of matching. Suppose that the centroid of O (or some other anchor point common to both O and M) is not known because O is partly obscured. Then we have to match the visible portion of O's surface against all possible similar portions of M's surface, and that may force us to iterate over (an appropriate discretization of) the 3-D rotational group R_3. This means that if we discretize R_3 into n^3 points, and for purpose of integration discretize S into n^2 points, we end up with an $O(n^5)$ matching procedure, far too slow to be useful. What is missing here is an appropriate generalization of the discrete fast Fourier transform algorithm to the case of (a discretized form of) the group O_3. Even so the complexity can be reduced to $O(n^4 \log n)$ by using the standard fast Fourier transform to handle all n members of O_3 which transform the north pole of S to the same point on S, all simultaneously. To do better than this, a generalization of the fast Fourier transform which can give some rapid way of evaluating integrals on the sphere is needed.

7. Numerical Experiments

The plausibility of the 2-D matching schemes suggested above can be confirmed by simple numerical simulations. For such simulations we begin by generating noisy star-shaped objects, representing hypothetical 'measurements'. These are generated by taking ideal convex objects, and perturbing some specified number of points on each of their sides by adding artificial randomized noise to their distance from the object centroid; this noise ranges between $1 - noise\_const$ and $1 + noise\_const$, where *noise_const* is a specifiable parameter controlling the amount of noise applied. A line of obscuration can also be specified for each simulation run, in which case all points of the polygon lying to the right of this line to be omitted from the simulated measurement. Each generated object O is then matched against a collection of ideal convex objects, including the one from which O has been generated by applying the above random perturbation.

Each of the two preceding heuristic matching schemes has been simulated. When the matching algorithm finds that two or more ideal objects have nearly the same shape-descriptor distance from a shape-descriptor it reports the error or ambiguity in specific terms.

Easy simulations have been run with the library of convex objects shown in the following figure. The results of these simple experiments are encouraging. Both matching schemes described identify the correct model object in almost all trials. Exceptions occur in cases when an observed object is obscured in a way which made its visible portion similar to a portion of another model object, or when the degree of random noise was high enough to confuse the measured object with a visually similar but different model object (e.g. a circle measured with 20 percent noise may be identified as an oval). The two matching heuristics yield similar results, but in the presence of large quantities of noise the second technique more reliably avoids the grotesque misidentifications that begin to plague the first method. The following figures are generated using the second matching scheme. Note that in each case the matching operation successfully identifies the figure presented to it, from among all the other figures belonging to the small shape library shown in Figure 7, using only those unobscured boundary portions indicated in Figures 8(c) (resp. 8(f)).[2]

---

[2] The authors would like to thank Charles Kim for assistance with the simulations described in this section.

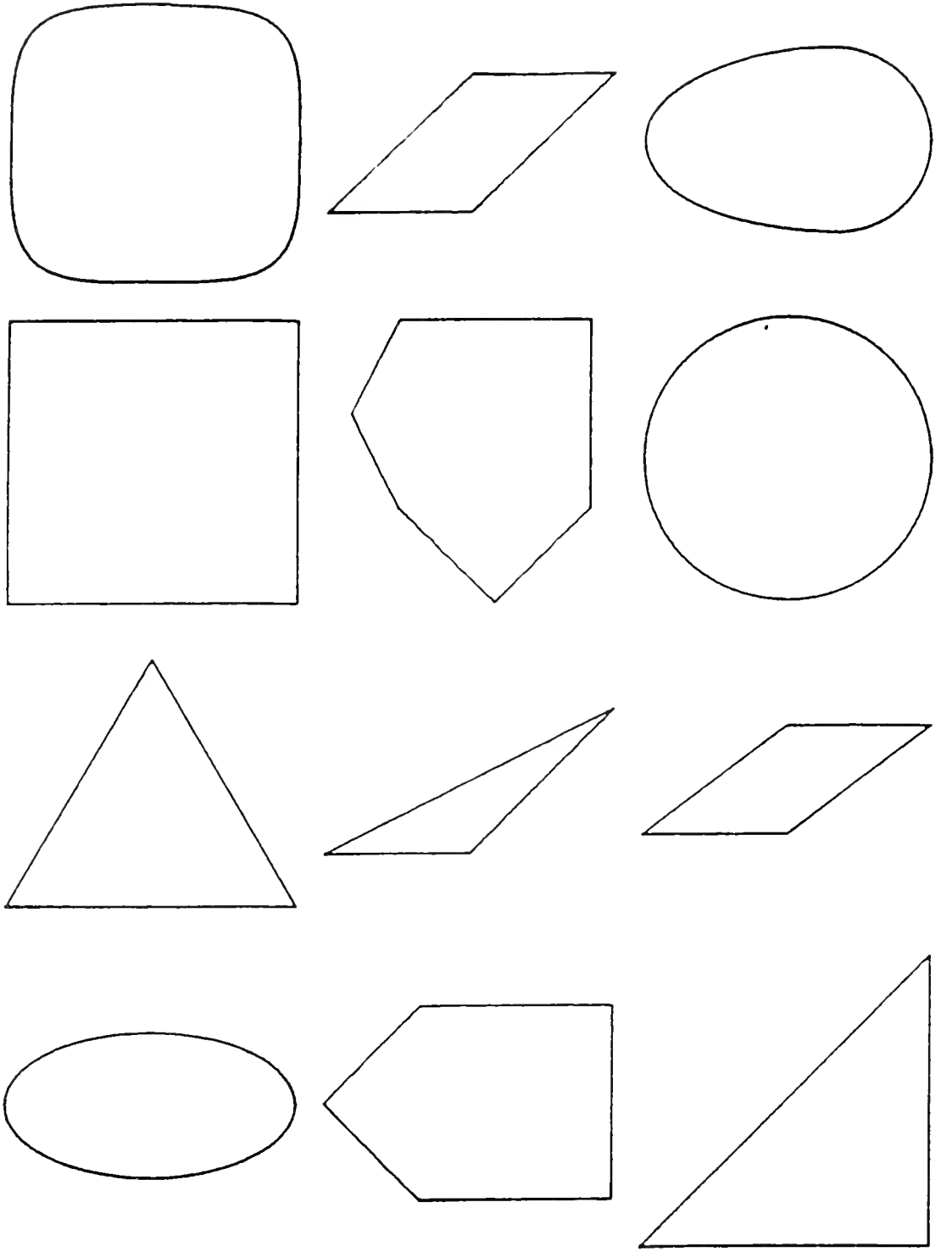

Figure 7: Various figures from library of test figures used in simulations of matching scheme

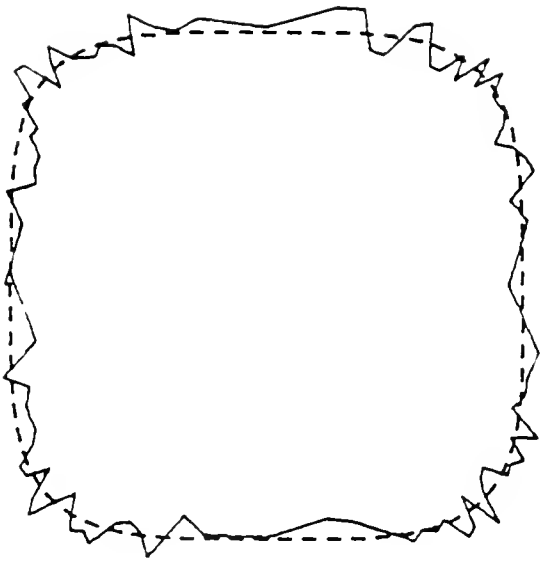

Figure 8(a): A test oval and its roughened form

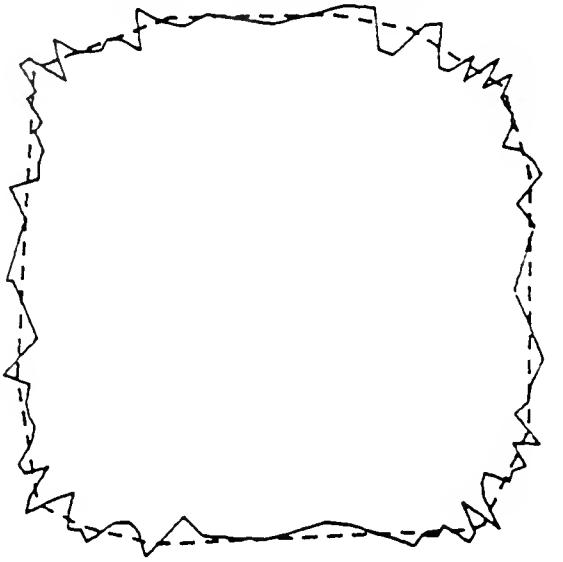

Figure 8(b): Adjusted convex hull of test oval

8. Additional remarks on the turning-angle shape descriptor; some remarks on texture; non-convex regions

Since the derivative of the rotated graph $\eta(s)$ derived from a region's 'turning function' $\theta(s)$ is bounded by 1 in modulus, the Fourier series of $\eta(s)$ converges to $\eta(s)$ with relative rapidity, making it possible to use the first few terms of this series as descriptors of the overall shape of the region. (The adequacy of such an

abbreviated description can be assessed by regenerating the figure from these Fourier coefficients, and then noting what differences with the original region the eye picks out.)

To judge the limitations of this abbreviation, polygon is regular, then the function $\theta(s)$ moves in alternate horizontal and vertical steps which must be equal in size (assuming normalization of the total arc length to 2π). Hence $\eta(s)$ is as shown in the following figure:

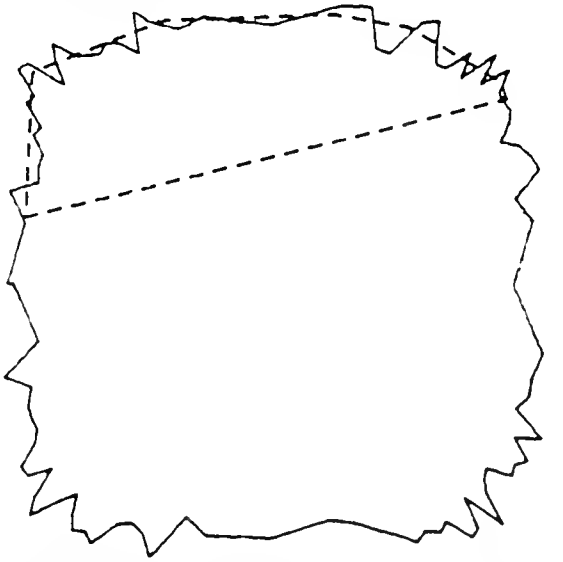

Figure 8(c): Visible portion of test oval hull

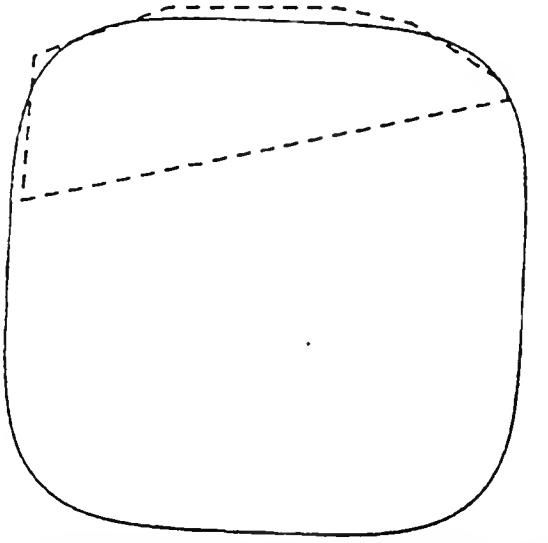

Figure 8(d): Complete match of oval to visible convex hull section

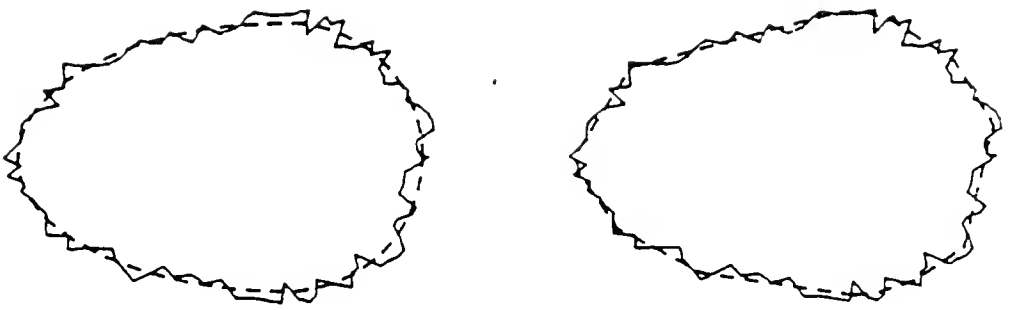

Figure 8(e): Original, roughened form, and adjusted convex hull of a second test oval

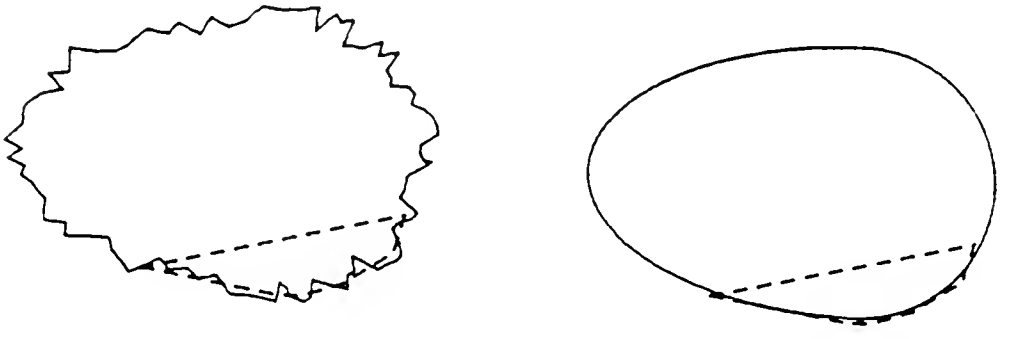

Figure 8(f): Visible portion and computer-generated match for second test oval

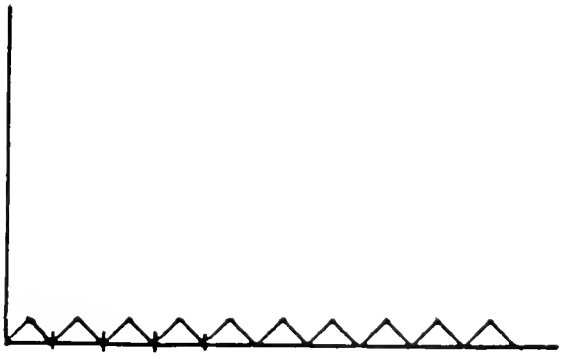

Figure 9: The function $\eta(s)$ for an n-sided regular polygon

A (unit) circle would have a perfectly flat $\eta(s)$ function. This comparison shows that the $\eta(s)$-function for an n-sided polygon approximates that for a circle very closely in the uniform norm, but that it has a significantly different geometric *texture*. To reflect this fact, a shape-matching scheme would have to apply some operator which will detect texture. The following is one possibiity: decompose the function η into parts corresponding to different frequency ranges by applying a disjoint set of band-pass filters. (These can decompose η into its low frequency part, encompassing all frequencies up to a limit F_1, and into then exponentially expanding frequency ranges F_1 to F_2, F_2 to F_3,... etc.) This gives $\eta(s) = \eta_1(s) + \eta_2(s) + \eta_3(s) + \cdots$, where relatively few terms need appear. The low-frequency component can be represented exactly by a few Fourier coefficients, after which each of the few higher-frequency components η_2, η_3, ... can be handled as follows: Calculate the variation of each such η_j over the range from 0 to s. This defines a monotone increasing function δ_j; treat this as previously, i.e. turn it 45° and represent it by its lowest few Fourier terms. The functions δ_j then represent the way that the texture of the original turning function $\theta(s)$ varies from zone to zone along the boundary of the region being analyzed. An approach like this might be able to represent the shape and texture of a convex body adequately using something like 25-35 numerical parameters: e.g. 4 sines and 5 cosines to represent each of η_1 and η_2, and 2 sines and 3 cosines for η_3, which will normally be much less significant to the eye.

As in the simpler case noted above, we can assess the adequacy of these descriptors by generating and inspecting the simplest curves which these descriptors fail to distinguish within a variety of examples presented to the analysis system. One obvious shortcoming is that the proposed scheme misses periodicities which the eye can pick out, e.g. it does not distinguish the function η_s appearing in the last preceding figure from the function which is different to the eye.

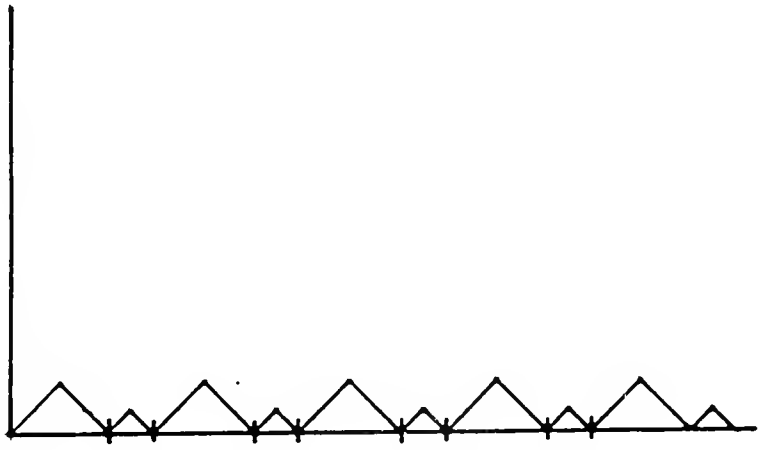

Figure 10: The function $\eta(s)$ for an n-sided but not entirely regular polygon

This remark may be more pessimistic than is justified, since we deal here with small edges on nearly circular polygons with numerous sides; nevertheless, only experiments exhibiting the strengths and weaknesses of the scheme proposed can establish the validity of more refined suggestions.

In concluding this section we note that *non-convex regions* turn out, somewhat surprisingly, to be easier to handle than convex regions. Every concavity is bounded by a single straight side of the convex hull of the body, which can be called the *entrance* to the concavity; if the region is partially obscured, the concavity can be said to have a *correctly visible entrance* if no point of obscuration lies on the opposite side of the entrance from the visible points of the region.

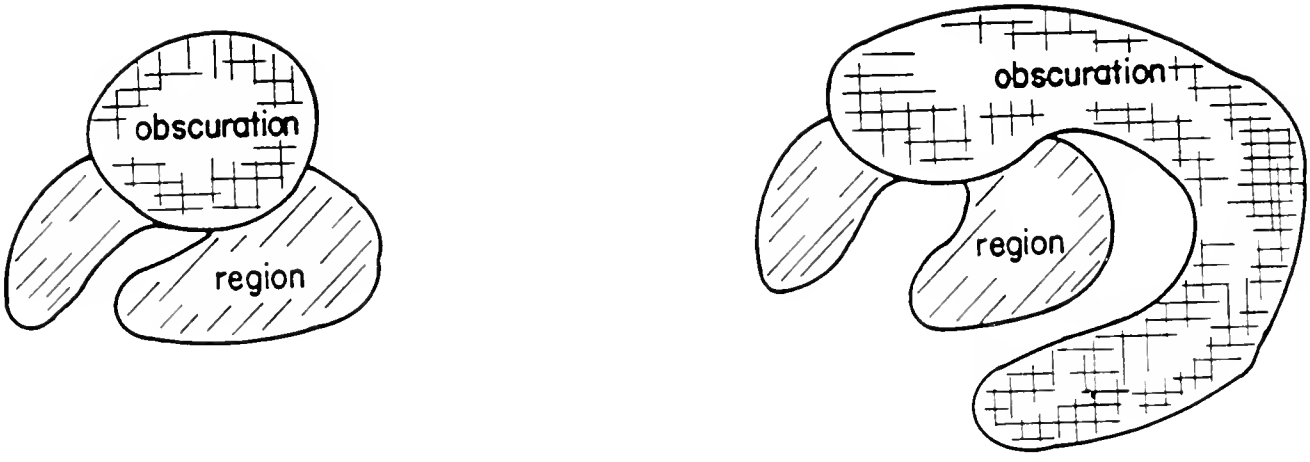

Figure 11: Concavity entrances (The left one is correctly visible, whereas the right one is not.)

A correctly visible concavity entrance identifies a straight side of the convex hull of the region to which it belongs, and since the hull has at least this one straight side we can treat it as 'partially polygonal', i.e. can take the polygon positions in which one such straight side is horizontal as the basis for the search tree used to identify all regions having a concavity with a correctly visible entrance. Moreover, any concavity with a correctly visible entrance can be considered to constitute a region (which may be partially obscured) in its own right; any technique applicable to (partially obscured) regions can be applied recursively to such a concavity. In particular, whenever the whole boundary of a concavity is visible, we can find its centroid and can use this as an anchor point for the region.

9. Three Dimensional Bodies

Having now said a good deal about the 2-D case, we can try to extend our considerations to the more challenging but more significant case of bodies in three dimensions. Let us begin by considering the probing technique, and first its application to the simplest case, that of a convex polyhedron standing on one of its faces. It will generally be easy to find two points p_1, p_2 fixed in the body, e.g. we can form silhouettes of the object as viewed from the x, y, and z directions and use this data to locate the topmost point and the point with largest x-coordinate (note that we need the 3-space locations of both these points). This will be determinable from the three silhouettess unless the topmost (resp. rightmost) edge or

face of the polyhedron is parallel to the xy (resp. yz) plane, in which case appropriate technical adjustments need to be made. Once these points have been found we can logically rotate and translate the body to put these two points in some standard position; then the object is necessarily in one of some finite number of possible positions (the number of these positions being roughly proportional to the number of faces of the polyhedron) and probes using a single-point depth sensor, organized in a 'probe tree' as before, should identify the polyhedron without difficulty.

A similar technique can be used even if the polyhedron does not necessarily stand on a known face. From three silhouettes, we can find the topmost and bottom-most points of the polyhedron, plus the points farthest left and farthest right. These define body position up to one of a finite number of fixed positions. (Similarly, if a horizontal face is topmost or bottommost, this fact, plus the location of the leftmost and rightmost points of the polyhedron, determines its position up to finitely many possibilities.) Then we can probe as before to complete identification and orientation of the polyhedron.

Neither of these techniques depends on polyhedron convexity. Indeed, either technique can be regarded as a way of orienting the convex hull of a non-convex polyhedron. Moreover, as soon as the position of its convex hull is limited to a finite set, the possible positions of the polyhedron itself become equally limited, and probing can be used in the ordinary way to complete its identification.

Next suppose that only part of a polyhedron is visible. If this part includes at least one corner, this corner and an edge running from it can be used to anchor the polyhedron, following which we can apply much the same probing technique as was described for partly obscured polygons.

To extend the probing idea to 3-D objects with smoothly curved boundaries, we need to find, not just one anchor point (as in the 2-D case), but two anchor points (or one anchor point and one 'anchor direction' emerging from it) which have known position relative to the object. Once these points are fixed, the object O is only free to turn about the axis defined by these two points, so that by probing along a circle perpendicular to this axis until contact is made with the body surface, we can restrict O's possible orientations to a finite set. After this is achieved, the probe tree method can be used to complete the identification of O. Note also that, once a single anchor point p for O has been located, finding a second anchor point q will generally reduce to a relatively easy 2-dimensional problem. If, for example, a sufficiently large portion of O's surface is visible, we can form the intersection of O with a sphere of appropriate radius about p, and let q be a point whose position in the resulting curve C is fixed; for example, q can be the centroid of C. When too little of O is visible for this approach to work, another possibility is to match C to an appropriate pre-stored model curve using the second of the fast matching techniques described in Section X.

10. 3-D Object Recognition by Shape Descriptors

To apply a shape-descriptor approach we must consider generalizations of the 2-D matching schemes presented above to the 3-D case. When the whole of O is visible, then an advantageous parametrization becomes possible. That is, we can take c to be the centroid of O, and parametrize the points on its boundary by the orientation of the ray connecting them to c. This parametrization is relatively immune to noise. Details are as follows. Let O be a convex 3-D object (if O is an observed object we assume it to be wholly visible). The shape descriptor that we want to use for O is simply a collection of data points $(\mathbf{u}(\omega))$ on its boundary, where each point $\mathbf{u}(\omega)$ is parametrized by the orientation ω of the ray connecting the centroid c of O with $\mathbf{u}$; in other words, this shape descriptor is a 3-D vector function defined on the unit sphere S (i.e. a generalized 'coloring' of the sphere).

If O is partially obscured its centroid cannot be determined, but instead we can use any other 'anchor point' having fixed location relative to the visible part of O's surface as the center for an angle-based parametrization. Once such a fixed parametrization of O's surface is available, the matching technique described previously may become available; but note the *caveats* concerning efficiency which have been expressed. Note also that the parametrization just outlined is also applicable to the case of nonconvex 3-D surfaces, provided that these surfaces are at least star-shaped with respect to some 'anchor point'.

11. Polyhedron Recognition Using Silhouettes

Given presently available sensors, object silhouettes can be formed more sharply and rapidly than depth images. For this reason, it is worth considering the extent in which the silhouettes of polyhedra can be used to identify them. To this end, the following remarks on silhouettes will be helpful. Suppose that a convex polyhedron P is given a certain orientation in 3-space, and projected upon a plane Q parallel to the xz plane which lies entirely on one side of P. Given any such orientation, one group of P's faces will be visible from Q, while its other faces will be obscured by the body of P. The boundary between the visible and the invisible portions is a sequence of edges of P, which we will call the *3-silhouette* of P; the projection onto the xz-plane of the 3-silhouette bounds the ordinary *2-D silhouette* of P, which is always a convex polygon. If we assume that no face of P is orthogonal to the xz-plane, then just one point p of the 3-silhouette projects onto each point q of its 2-D silhouette, and p varies continuously with q. Hence the 3-silhouette is topologically a circle, and therefore divides the surface of P into exactly 2 groups of faces, each of which must be connected. The silhouette of a convex polyhedron P is therefore the projection, on the camera's image plane, of a closed sequence of edges on P.

Suppose we draw the outward-directed normal $\mathbf{n}$ to a given face F of P. Then F is visible from Q if $\mathbf{n}$ points toward Q, but obscured by the body of P if $\mathbf{n}$ points away from Q. To understand how the 3-silhouette of P varies as we rotate P about a vertical axis, it is convenient to project all the normals $\mathbf{n}$ to P's faces F

onto the xy plane. This forms a 'direction diagram' consisting of unit vectors in the xy plane, and then a face is visible from Z if the corresponding projected normal points toward one side, say the negative side, of the y axis, but is invisible otherwise. Thus the edge separating two adjacent faces F_1 and F_2 belongs to P's 3-silhouette if and only if the projected normal vectors to F1 and F2 point into opposite sides of the y axis.

Take an edge of the silhouette and its two extremities u_1, u_2. These are projections of corresponding polyhedron vertices v_1, v_2, which therefore lie along two known lines in space. Taking the eye of the camera to be the origin of coordinates, we can therefore write $v_1 = x\mathbf{a}_1$, $v_2 = x\mathbf{a}_2$, where $\mathbf{a}_1$, $\mathbf{a}_2$ are known unit vectors. Let u_2u_3 be the next edge of the silhouette. Ignoring exceptional positions, this must correspond to a polyhedron edge v_2v_3, and again we have $v_3 = z\mathbf{a}_3$ where $\mathbf{a}_3$ is a known unit vector. It will be noted below that the maximal number of distinct 3-silhouettes that P can have is $O(n^3)$ ($O(n^2)$ if we consider only *isometric* silhouettes), and in fact the maximal number of distinct triplets v_1, v_2, v_3 of adjacent vertices of P in a silhouette is also at most $O(n^3)$ ($O(n^2)$ in the isometric case). Hence (searching as always over a finite number of possibilities) we can suppose that the three distances $D_1 = |v_1v_2|$, $D_2 = |v_2v_3|$, $D_3 = |v_3v_1|$ are known. This gives us three quadratic equations for determining the three unknowns x, y, z, namely

$$x^2 + y^2 - 2xy(v_1 \cdot v_2) = D_1^2$$

$$y^2 + z^2 - 2yz(v_2 \cdot v_3) = D_2^2$$

$$z^2 + x^2 - 2xz(v_3 \cdot v_1) = D_3^2$$

These can readily be solved by subtracting suitable multiples of the third equation from the first two, which gives two inhomogeneous quadratic equations for the ratios $\xi = x/z$ and $\eta = y/z$. Thus knowing the positions of these successive vertices on the perimeter of the silhouette determines the polyhedron orientation up to a finite number of possibilities, and hence determines the entire silhouette in the same sense. If the silhouette has four or more vertices we should therefore be able to compare a finite collection of calculated silhouettes with an actual silhouette, and this will often identify the body and its orientation uniquely. Identification becomes even easier if we assume that silhouettes viewed from two slightly different angles are available; we leave it to the reader to work out the details involved.

As usual, the situation is somewhat more favorable if the polyhedron and its silhouette are nonconvex. In such case each vertex of the silhouette which has the property that the silhouette lies on the smaller side of the two silhouette edges forming the silhouette boundary ('convex corners') must correspond to a corner of the polyhedron. Moreover, any segment connecting two such points which does not form part of the silhouette boundary must be the image of an edge ('flying

edge') which connects two corners of the polyhedron but is not an edge of the polyhedron. Often this observation will make it possible to identify silhouette vertices rapidly. Consider, for example, the union of parallelipiped and pyramid shown in the following figure:

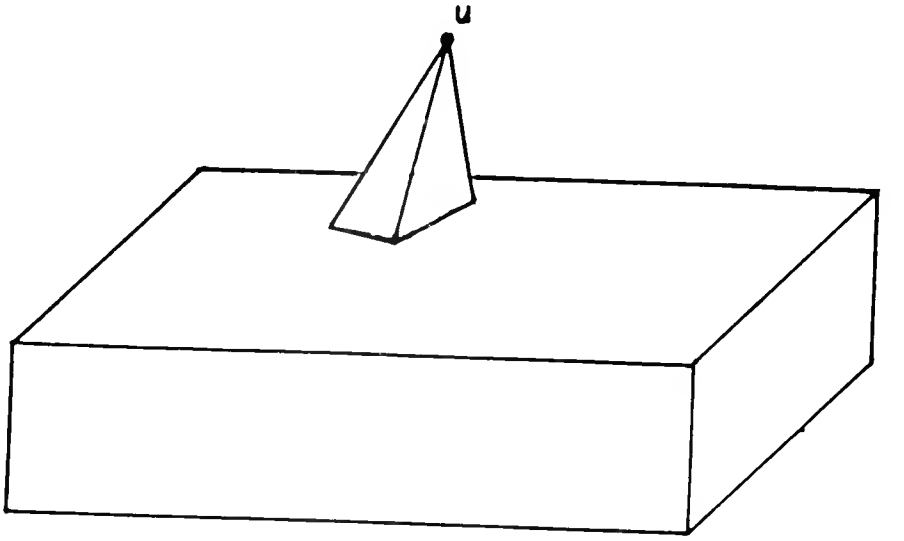

Figure 12: A non-convex polyhedral object

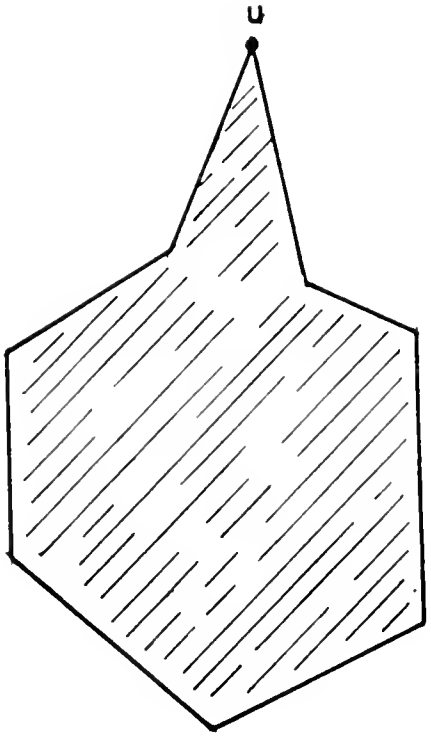

Figure 13: Silhouette of a non-convex polyhedral object

In the silhouette the point u must correpond to v since v is the only polyhedron vertex connected to more than one other vertex by a flying edge.

To show that for any convex polyhedron P there exist at most $O(n^3)$ distinct silhouettes and, in the case of isometric projections, only $O(n^2)$ silhouettes, we can argue as follows. Consider the isometric case first. An isometric silhouette is uniquely determined by the direction in which P is projected, and hence in turn by a point on the unit sphere. As already noted, a silhouette changes its combinatorial structure only when it is projected in a direction $\mathbf{v}$ at which some face F of P is seen end-on; in other words, only when $\mathbf{v}$ is perpendicular to the normal $\mathbf{n}_F$ of F. But for each face F, the locus of orientations $\mathbf{v}$ which satisfy this condition is a great circle on the surface of the unit sphere. Since P has n faces, this defines n such great circles which collectively partition the sphere into $O(n^2)$ open regions, inside each of which the combinatorial structure of P's silhouette remains constant.

Next consider silhouettes seen from an arbitrary viewpoint Z. In this case the combinatorial structure of a silhouette can change only when Z lies on one of the face planes of P. These n planes decompose the 3-D space exterior to P into

$O(n^3)$ regions, structure of the silhouette remains const

[IS82] K. I Recognition System for Recogni;tion Conf. on Artificial Intelligence (1982),

[OS75] M. (of Curved Objects Using Three-Dimens Computer Conf. (1975), pp. 108-112.

[OS79] M. C ion Method Using Three-Dimensional I 979), pp. 9-17.

[Sc83] J.T. S Robot Vision, New York University Te o.8, 1983.

[S79] Y. Shi , in Computer Vision and Sensor-Based Plenum Press, N.Y., 1979, pp. 187-205

[SKOI83] Y. Shirai, K. Koshikawa, M. Oshima, and K. Ikeuchi, A Vision System Based on Three-Dimensional Model, Proc. 1983 Int. Conf. on Advanced Robotics (ICAR83), Tokyo, pp. 139-146.

[SS71] Y. Shirai and M. Suwa, Recognition of Polyhedrons with a Range Finder, Proc. 2nd Int. Joint Conf. on Artificial Intelligence (1971), pp. 80-87.

[T83] Technical Arts Corporation, The White Scanner 100 Series, Technical and Sales Brochures, 1983, (Address: 100 Nickerson, Suite j102, Seattle, WA 48109).